Remarks

Reconsideration of the application is respectfully requested in view of the foregoing amendments and following remarks. Claims 1-38 remain in the application.

Objections to Specification and Claims

The Office asserted various informal objections to the Brief Description of the Drawings and to claims 8, 13, 24, 31, and 32. *See* Office Action, mailed September 22, 2004, ¶¶ 1-8, pages 1-3. Applicants have made all of the informal amendments or modifications discussed by the Examiner. The specification and claims should now be in condition for allowance. Such action is respectfully requested.

Claim Definiteness

The Office asserted a rejection of claims 1-27, 31-33, and 35-38 under 35 U.S.C. § 112, second paragraph, because they are said to be indefinite for failing to distinctly claim the subject matter of the invention. *See* Office Action, mailed September 22, 2004, ¶¶ 9-18, pages 3-5. Applicants have made all of the modifications discussed by the Examiner. The specification and claims should now be in condition for allowance. Such action is respectfully requested.

Patentability Over DeLong or Lindholm

The Office has asserted a novelty rejection of claims 1-14, 19-22, 25, 27-28, and 30-38, under 35 U.S.C. § 102(e) over U.S. Patent No. 6,247,169 ("DeLong"). Additionally, a novelty rejection has been asserted against claims 15, 17, 26 and 29 over U.S. Patent No. 6,618,855 ("Lindholm"). Respectfully, Applicants traverse these rejections.

Claim 1

Applicants respectfully submit that the Office has failed to establish anticipation, because DeLong fails to teach or suggest "placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream." Specifically, claim 1, with emphasis, recites,

> 1.   (currently amended) A method of translating computer program
> code from a first language representation into a second language representation,
> the method comprising:

translating ~~the~~ translatable instructions of an input stream in a first
language representation into an output stream in a second language
representation;

identifying an unresolvable translation error in the input stream; and

**placing at least one second language representation instruction in the
output stream responsive to identifying the unresolvable translation error in
the input stream;**

wherein the placed at least one second language representation instruction
is at least one of either a handling instruction or an exception throwing
instruction.

Applicants respectfully submit that DeLong fails to teach or suggest "placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream." For example, DeLong fails to teach anything about "unresolvable translation errors in the input stream." As stated in the present Application, unresolvable translation errors occur, for example, during syntax translation, semantic translation, lexical translation, logical translation, code verification translation, or type check translation. *See e.g.,* pages 10-12. When a translator encounters such unresolvable code in the input stream, the translator is unable to translate the unresolvable code. *See e.g.,* page 12, lines 23-24; page 19, lines 1-9. Applicants respectfully submit that the art of record in the present application fails to teach or suggest anything about an "unresolvable translation error in the input stream."

Exception handling for protected code that has already been translated teaches away from the recited arrangement

Without first showing such "an unresolvable translation error in the input stream" it is impossible to show "placing at least one second language representation instruction in the output stream" responsive thereto. As stated in the present Application, at page 5, lines 14-16,

Unresolvable instructions are not translated since their intended meaning
is not known, so they are instead converted into an exception-throwing
instruction(s) (and/or handling instructions) and inserted or placed into the output
code.

For example, DeLong merely discusses a method for encapsulating a protected region of code within an exception construct. *See e.g.,* DeLong, the Abstract. This is evident in that DeLong discusses that the protected region of code can be "run" by the processor. *See e.g.,* DeLong, col. 3, line 30. But nowhere does DeLong teach or suggest anything about an "unresolvable translation error in the input stream."

DeLong is merely a description of prior art method for exception handling for "protected code" that has *already been translated*. Thus, DeLong is describing language constructs that are requesting exception handling for certain behavior that occurs in already translated "protected code." Thus, Delong assumes the "protected code" is already translated (e.g., compiled) and can now be run.

For example, DeLong assumes that the protected code is already translated into an executable form (col. 5, line 5) and an exception might not be raised at all during execution of the protected region (col. 5, lines 7-8). Thus, the protected region is assumed to be already translated into an executable form. Thus, DeLong teaches squarely away from "unresolvable translation error in the input stream."

For at least this reason claim 1 is allowable. Such action is respectfully requested.

Claims 2-14, 19, 20, 33, and 35-38

Claims 2-14, 19, 20, 33 and 35-38 depend from claim 1. Since they depend from claim 1, they should be allowed for at least the reasons stated for claim 1. In view of the foregoing discussion of claim 1, the merits of the separate patentability of dependent claims 2-14, 19, 20, 33 and 35-38 are not belabored at this time. Claims 2-14, 19, 20, 33 and 35-38 should be allowable. Such action is respectfully requested.

Independent Claims 21, 25, 27, 28, 30, 34

Applicants respectfully submit that for reasons similar to those stated above, such as for claim 1, DeLong fails to teach or suggest the following features:

Claim 21—"identifying a first language representation of a declarative textual indication in the input stream, the declarative textual indication indicating how to handle an unresolvable translation error encountered in the input stream"

Claim 25—" instruction(s) placing an exception throwing instruction in the second language output stream in response to identifying the unresolvable translation error in the first language input stream"

Claim 27—" instruction(s) placing a handling instruction in the second language output stream in response to identifying the unresolvable translation error in the first language input stream"

Claim 28—"a translation sub-unit operational to translate an unresolvable translation error identified in the first language input stream into an inserted exception throwing instruction in a translated code of the second language output stream"

Claim 30—"a translation sub-unit operational to translate an unresolvable translation error encountered in the first language input stream into an inserted handling instruction in a translated code of the second language output stream"

Claim 34—"inserting the second language representation of the exception throwing instruction into the second file where the identified unresolvable instruction would have been placed in the second file by the compiler if the unresolvable instruction were resolvable"

Since DeLong fails to teach or suggest these features of independent claims 21, 25, 27, 28, 30 and 34 they should be allowable. Such action is respectfully requested.

Dependent Claims 22, 23, 31, and 32

Claims 22, 23, 31, and 32 depend from the above allowable independent claims. Since claims 22, 23, 31, and 32 depend from the above allowable independent claims, they should be allowed for at least the above reasons. Such action is respectfully requested.

Claim 15

Applicants respectfully submit that the Office has failed to establish anticipation, because Lindholm fails to teach or suggest a "placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream."

Specifically, claim 15, with emphasis, recites,

> 15. (currently amended) A method of translating computer program code from an input stream in a first language representation into an output stream in a second language representation, and the input stream may or may not be from a trusted source, the method comprising:
> translating translatable instructions of the input stream into the output stream;
> identifying suspected code and unresolvable translation errors in the input stream;
> **placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream;**
> determining that the input stream is from a trusted source; and
> translating the suspected code in the input stream into the output stream.

Applicants respectfully submit that Lindholm fails to teach or suggest "placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream."

For example, Lindholm states that verification "insures that illegal operations are not attempted ... that can lead to meaningless results or that can compromise the integrity of the operating system." *See e.g.,* col. 4 lines 37-40. Further, Lindholm states that when a "Java language program violates the constraints of an operation, the JVM detects an invalid condition and signals this error to the program as an exception." *See e.g.,* col. 3, lines 58-60. However, this is merely another example of the translator cannon—when a translation error is detected—abort translation.

Aborting translation and throwing an exception when a class cannot be verified teaches away from the recited arrangement

Thus, Lindholm is describing the prior art method of aborting translation and throwing an exception thereby ensuring "an illegal operation is not attempted." *See e.g.,* Lindholm, col. 4 lines 37-40. This merely restates the discussion of the prior art as discussed in the present Application, at page 3, lines 24-29, and page 10, lines 22-29,

> When a sub-unit of translation aborts translation because it is unable to resolve the meaning of an instruction(s), then that instruction is deemed an unresolvable instruction(s) (or unresolvable code). (See below Unresolvable Translation Errors). If the translator is unable to resolve the translation in a meaningful or unambiguous way, then translation aborts.
>
> ...
>
> As shown in Figure 2, what all these translators 200 have in common is a potential for identifying instructions, data, or objects 204 in the input stream (in this specification "input file" has the same effective meaning as "input stream" and "output file" has the same effective meaning as "output stream") of the first language that for one reason or another the translator is unable to unambiguously, meaningfully, or safely translate into corresponding instructions, data, or objects in the second language 202, 208. In such a case, the translator typically will abort translation and deliver a message 206 to an output device (screen, file or printer) 203 identifying the source of the unresolvable instruction(s), data, or objects in the input stream 204.

Applicants respectfully submit that aborting translation and throwing an exception when a class can't be verified teaches away from the recited arrangement. Thus, Lindholm fails to teach or suggest "placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream."

For at least this reason claim 15 is allowable. Such action is respectfully requested.

Claim 17

Claim 17 depends from claim 15. Since claim 17 depends from claim 1, it should be allowed for at least the reasons stated for claim 1. In view of the foregoing discussion of claim 1, the merits of the separate patentability of dependent claim 17 is not belabored at this time. Claim 17 should be allowable. Such action is respectfully requested.

Independent Claims 26, 29

Applicants respectfully submit that for reasons similar to those stated above, such as for claim 1, Lindholm fails to teach or suggest the following features:

Claim 26—"instruction(s) placing an exception throwing instruction in the second language output stream in response to identifying the unresolvable translation error in the first language input stream"

Claim 29—"a translation sub-unit operational to translate an unresolvable translation error encountered in the first language input stream into an inserted handling instruction in a translated code of the second language output stream"

Since Lindholm fails to teach or suggest these features of independent claims 26 and 29 they should be allowable. Such action is respectfully requested.

Patentability Over DeLong and Lindholm

The Office has asserted a rejection of claim 18 under 35 U.S.C. § 103(a) over DeLong in view of one of ordinary skill in the art. Additionally, an obviousness rejection has been asserted against claim 16 over Lindholm in view of DeLong, and against claims 23 and 24 over DeLong in view of Lindholm. Respectfully, Applicants traverse these rejections.

Claim 18

Claim 18 depends on claim 1. Applicants respectfully assert that the Office has failed to carry the burden of establishing a prima facie case of obviousness of claim 1, because the art of record fails to teach or suggest "placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream."

For example, DeLong merely discusses a method for encapsulating a protected region of code within an exception construct. *See e.g.,* DeLong, the Abstract. This is evident in that DeLong

discusses that the protected region of code can be "run" by the processor. *See e.g.,* DeLong, col. 3, line 30. But nowhere does DeLong teach or suggest anything about an "unresolvable translation error in the input stream." For example, DeLong assumes that the protected code is already translated into an executable form (col. 5, line 5) and an exception might not be raised at all during execution of the protected region (col. 5, lines 7-8). Thus, the protected region is assumed to be already translated into an executable form. Thus, DeLong teaches squarely away from "unresolvable translation error in the input stream."

Additionally, the Office asserts that "requesting by the presently executing at least one placed second language representation instruction, the request being made to a server for instructions external to the output stream" is "common practice." Applicants respectfully request a reference supporting the Office's assertion of common practice or knowledge. MPEP § 2144.04.

However, since claim 18 depends on allowable claim 1, it should now be allowable since the Office has failed to teach or suggest "placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream." For at least this reason claim 18 is in condition for allowance. Such action is respectfully requested.

Claim 16

Claim 16 depends on claim 15. Applicants respectfully assert that the Office has failed to carry the burden of establishing a prima facie case of obviousness of claim 15, because the art of record fails to teach or suggest "placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream."

For example, Lindholm states that verification "insures that illegal operations are not attempted ... that can lead to meaningless results or that can compromise the integrity of the operating system." *See e.g.,* col. 4 lines 37-40. Further, Lindholm states that when a "Java language program violates the constraints of an operation, the JVM detects an invalid condition and signals this error to the program as an exception." *See e.g.,* col. 3, lines 58-60. However, this is merely another example of the translator cannon—when a translation error is detected—abort translation. Applicants respectfully submit that aborting translation and throwing an exception when a class can't be verified teaches away from the recited arrangement. Thus, Lindholm fails to teach or suggest "placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream."

Further, Delong fails to teach or suggest "placing at least one second language representation instruction in the output stream responsive to identifying the unresolvable translation error in the input stream." For example, DeLong merely discusses a method for encapsulating a protected region of code within an exception construct. *See e.g.,* DeLong, the Abstract. This is evident in that DeLong discusses that the protected region of code can be "run" by the processor. *See e.g.,* DeLong, col. 3, line 30. But nowhere does DeLong teach or suggest anything about an "unresolvable translation error in the input stream." For example, DeLong assumes that the protected code is already translated into an executable form (col. 5, line 5) and an exception might not be raised at all during execution of the protected region (col. 5, lines 7-8). Thus, the protected region is assumed to be already translated into an executable form. Thus, DeLong teaches squarely away from "unresolvable translation error in the input stream."

Thus, no combination of Lindholm and DeLong describes a method that places a "second language representation instruction in the output stream" responsive to identifying an "unresolvable translation error in the input stream." The art of record in this application simply fails to teach or suggest this arrangement. For at least this reason claim 15 and its dependent claim 16 are in condition for allowance. Such action is respectfully requested.

Claim 23 and 24

Applicants respectfully submit that for reasons similar to those stated above for claim 16, the DeLong-Lindholm combination fails to teach or suggest the following features:

Claim 21—"identifying a first language representation of a declarative textual indication in the input stream, the declarative textual indication indicating how to handle an unresolvable translation error encountered in the input stream"

Claim 24—"identifying a second unresolvable translation error in a second method in the input stream, determining by the basic block identification sub-unit of translation, that the second unresolvable translation error can be safely segregated within a determinable basic block within the second method, and invoking a basic block level exception throwing instruction insertion sub-unit of translation to insert an exception throwing instruction in the output stream"
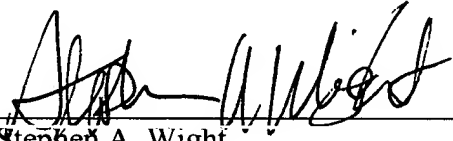
For at least this reason claim 21 and its dependent claim 23 are in condition for allowance. Such action is respectfully requested. Additionally, for at least this reason claim 24 is in condition for allowance. Such action is respectfully requested.

<u>Conclusion</u>

The claims in their present form should now be allowable. Such action is respectfully requested.

Respectfully submitted,

KLARQUIST SPARKMAN, LLP

By _____
Stephen A. Wight
Registration No. 37,759

One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Telephone: (503) 226-7391
Facsimile: (503) 228-9446